

REMARKS/ARGUMENTS

Prior to this amendment, claims 1-26 were pending. In this amendment, claim 19 is amended, no claims are canceled and claim 27 is added. No new matter is added. Thus, after entry of this amendment, claims 1-27 are pending.

I. Claim Rejections – 35 USC § 103(a), Drake, Pastor, Sang-Kyun,

Claims 1, 3, 5, 6, 14, 15, 17, 19 and 25-26 are rejected under 35 U.S.C. 103(a) as being obvious over *Drake* (U.S. Publication No. 2003/0070142), in view of *Pastor* (U.S. Patent No. 6,681,383), and further in view of Sang-Kyun Kim (“*Sang*”) (“Immediate and Partial Validation Mechanism for the Conflict Resolution of Update Operations in XML Databases.”). Applicants respectfully submit that these references do not teach or suggest each element of these claims, as presented or amended. This rejection is respectfully traversed.

Claim 1

Claim 1 is not amended. Applicants’ respectfully request that any new rejection of claim 1 be made non-final as it would not have been necessitated by applicants’ amendment. Each and every limitation of claim 1 is not taught or suggested by the cited references, alone or in combination.

For example, claim 1 recites in part, “*applying the validation rules to each object in the list of objects.*” The rejection of claim 1 does not address this limitation entirely and has failed to make a *prima facie* case of obviousness. Instead, the Office Action addresses the limitation “applying the determined validation rules to each subject in the context,” which was present in the claims as originally filed but is not present in the claim as examined in the instant Office Action. (Office Action, Pg. 4). As such, the Office Action has failed to make a *prima facie* case of obviousness with respect to the limitation, “*applying the validation rules to each object in the list of objects.*”

As another example, claim 1 recites in part, “*querying the database to retrieve one or more validation rules for each object in the list of objects.*” The Office Action alleges such a

limitation is disclosed by *Drake*, Fig. 3, P[0011, 0019-0020, and 0024-0025]. (Office Action Pg. 4). *Drake* does not disclose or suggest “*querying the database to retrieve one or more validation rules for each object in the list of objects,*” and actually describes the opposite, wherein validation rules are encapsulated within data objects. For example, P[0018] of *Drake*, reproduced below recites that validation is coupled with or encapsulated with the data model itself.

[0018] The present invention defines novel techniques for performing data validation. Validation is coupled with, or encapsulated with, the data values to which the data validation pertains, thereby becoming a part of the data model itself. This approach enables real-time data validation, as a user interacts with a data model through an executing application or GUI window interface.

As such, if the validation is coupled with the data model itself, there would be no need, nor is it suggested by *Drake*, to query a database to retrieve validation rules. A similar teaching is presented in P[0011] of *Drake* which is relied on by the Office Action, reproduced in part below, which also teaches that the validation is encapsulated with the data model.

[0011] To achieve the foregoing objects, and in accordance with the purpose of the invention as broadly described herein, the present invention provides methods, systems, and computer program products for improving data validation. In one aspect, this technique comprises defining one or more validation criteria and encapsulating the defined validation criteria with a data model to which they apply. The technique may also comprise using the defined validation criteria to

P[0019-0020] and P[0024-0025] of *Drake* also contain language similar to that above, in which the validation rules are coupled with the data, forming a complex object. (*Drake*, P[0019]). As such, *Drake* describes the validation rules being encapsulated within the data object, and as such, there would be no need to query a database to retrieve validation rules. *Drake* does not disclose

or suggest “*querying the database to retrieve one or more validation rules for each object in the list of objects.*” In attempting to make a *prima facie* case of obviousness, the Office Action alleges that such a limitation is disclosed in *Drake*. The Office Action has not alleged that a combination of the references would result in such a limitation, and has failed to make a *prima facie* case that the combination of the cited references would result in such a limitation.

As yet another example, claim 1 recites in part, “*querying the database to retrieve a list of objects requiring validation, the list of objects determined by the first subject of validation and the determined context.*” The limitation recites that the list of objects is determined by both the first subject of validation as well as the context. The Office Action alleges this is disclosed in *Sang*, P(388-390), wherein partial and full validation of an XML document is described. However, as acknowledged by *Sang*, a full validation of the XML document verifies every element in the XML document. (*Sang*, P(389)). As such, the list of objects to retrieve for a full validation is determined solely by the context, not by the “*first subject of validation and the determined context.*” In the case of a partial validation, the objects to be validated cannot be retrieved from a database query, because at minimum, the “*first subject of validation,*” (e.g. the data element to be inserted into the XML document) does not yet exist in the XML document. *Sang*, P(389) recites:

Most of XML database systems use *deferred and full validation* method when XML documents are updated. As we described in section 2, this method has conflicts and performance problems for update operations. Therefore, for solving these problems, XML database systems must be able to support *immediate and partial validation* method. This may immediately validate for the update operations before the actual update operation is applied to the XML document in the database and is also of higher performance compared to other systems for validation operations because it is able to validate only modified parts of a document. The validation processes to support immediate and partial validation are as follows:

As stated by *Sang*, partial validation occurs before the update operation if performed on the XML document in the database. As such, it would not be possible for *Sang* to disclose or suggest “*querying the database to retrieve a list of objects requiring validation, the list of objects determined by the first subject of validation and the determined context,*” because in a partial validation, the objects to be validated (e.g. the subject of validation) has not yet been stored in

the XML document in the database. It is not possible to retrieve an item from a database if the item has not been stored in the database.

Withdrawal of the rejection of claim 1 and the claims which depend therefrom is respectfully requested for at least the reasons set forth above.

Claim 17

Claim 17 contains limitations that are not taught or suggested by the cited references for reasons including at least those set forth with respect to claim 1. Withdrawal of the rejection of claim 17, and the claims which depend therefrom, is respectfully requested.

Claim 19

Claim 19 contains limitations that are not taught or suggested by the cited references for reasons including at least those set forth with respect to claim 1. In the interests of advancing prosecution, claim 19 has been amended to further clarify the distinctions over the current art of record.

For example, claim 19, as amended, recites in part:

receiving a selection from the user of a type of validation to perform, the type selected from the group consisting of correctness and completeness;

applying a correctness type validation rule to each object in the list of objects when the selection indicates correctness type validation; and applying a correctness type and a completeness type validation rule to each object in the list of objects when the selection indicates completeness type validation;

(emphasis added). None of the cited references teaches or suggests such a limitation.

Drake teaches coupling validation with the relevant data values such that the validation becomes part of the data model (paragraph [0018]). *Drake* further teaches that the validation process can be done at an early point, such as where the data model is preloaded with data values (paragraph [0021]). In such a case, a complete validation is done with pre-populated data, which can be altered/mutated and another validation done (paragraph [0021]). *Drake* does not disclose or suggest receiving a user selection of the type of validation to perform. In fact, there is only

one type of validation described in *Drake*, which is correctness (e.g. is the social security number formatted correctly). *Drake* does not describe a completeness type validation at all (e.g. is other data besides a social security number needed?).

Pastor describes a system for generating software from a well defined formal specification. *Pastor* may describe completeness and correctness validation (*Pastor*, column 20-23), however *Pastor* does not describe receiving a user selection for the type of validation to perform.

Sang describes a system for validation of XML documents. (*Sang*, Abstract). The Office Action equates partial or full validation with correctness and completeness validation. (Office Action, Pg. 12). However, under such an interpretation, *Sang* does not teach or suggest receiving a user selection for the type of validation. Rather, *Sang* describes a system wherein full validation (e.g. completeness) is replaced by partial validation (correctness). (*Sang*, Pg. 389). *Sang* does not describe allowing a user to select which type of validation is performed.

The addition of *Mikhailov* (US 6,968,500) or *Lindberg* (US 2003/0028540) does not resolve the above mentioned deficiency. None of the cited references individually teach or suggest the limitations added to claim 19. The combination of the references would likewise not teach or suggest the limitations of amended claim 19, as the combination would not inherently result in such limitations being taught or suggested.

Allowance of claim 19 and the claims which depend therefrom is respectfully requested.

II. Claim Rejections under 35 USC § 103, Drake, Rasmussen, Sang-Kyun, Mikhailov

Claims 2, 4, 18, and 20-21 are rejected under 35 U.S.C. §103(a) as being obvious over *Drake* and *Pastor* and *Sang* further in view of *Mikhailov* (US 6,968,500). These claims are not rendered obvious by *Drake* and *Pastor* and *Sang* as discussed above. Combining *Mikhailov* with these references, even if there were motivation to do so, still would not render these claims obvious. *Mikhailov* teaches an automatic forms handling system (col. 1, lines 8-15; col. 5, lines 19-38), and is cited as teaching a group of types of associated metadata (Office Action, pg. 14).

A combination of these references still would not result in, or provide motivation for the limitations as discussed above. Withdrawal of this rejection is respectfully requested.

III. Claim Rejections under 35 USC § 103, Drake, Rasmussen, Sang-Kyun, Lindberg

Claims 7-13, 16, and 22-24 are rejected under 35 U.S.C. §103(a) as being obvious over *Drake* and *Pastor* and *Sang* further in view of *Lindberg* (US 2003/0028540). These claims are not rendered obvious by *Drake* and *Pastor* and *Sang* as discussed above. Combining *Lindberg* with these references, even if there were motivation to do so, still would not render these claims obvious. *Lindberg* teaches a system for transferring information between a user interface and a database over a network (paragraph [0010]), and is cited as teaching a first subject as a root object for a collection of associated objects (Office Action, pg. 16). A combination of these references still would not result in, or provide motivation for the limitations as discussed above. Withdrawal of this rejection is respectfully requested.

IV. New Claims

Claim 27 is newly added. Claim 27 is a computer readable medium claim that contains limitations that are similar to claim 19. Support for claim 27 can be found throughout the specification, including places such as P[0037]. Claim 27 is allowable for at least the reasons as set forth above with respect to claim 19.

V. Amendment to the Claims

Unless otherwise specified or addressed in the remarks section, amendments to the claims are made for purposes of clarity, and are not intended to alter the scope of the claims or limit any equivalents thereof. Support for the amendments can be found throughout the specification, including places such as P[0029].

CONCLUSION

In view of the foregoing, Applicants believe all claims now pending in this Application are in condition for allowance. The issuance of a formal Notice of Allowance at an early date is respectfully requested.

If the Examiner believes a telephone conference would expedite prosecution of this application, please telephone the undersigned at 415-576-0200.

Respectfully submitted,

/Preetam B. Pagar/

Preetam B. Pagar
Reg. No. 57,684

TOWNSEND and TOWNSEND and CREW LLP
Two Embarcadero Center, Eighth Floor
San Francisco, California 94111-3834
Tel: 415-576-0200 / Fax: 415-576-0300
PBP:scz
62228683 v1